NAME

vw – viewer for smv

SYNOPSIS

vw [file.[smv,v]]

DESCRIPTION

This is a graphical user interface for *smv(1).* Provides source browsing, counterexample browsing, abstraction editing, proof management, and spreadsheet-style simulation. The file name is optional, and can be specified instead by pulling down the "File" menu and choosing "Open...". All input files are piped through a preprocessor much like the C preprocessor. Files with extension ".v", are interpreted as Synchonous Verilog, and are translated to SMV by *vl2smv(1).*

There are no command line switches, as in *smv(1).* Any smv compile-time options, such as -D and -I should be put in the file "file.cflags". See *smv(1)* for definitions of these switches. This file may also contain global options that apply to all properties to be verified. If there is no "file.cflags", some default options are used (see *smv(1)*). Model checking options that are generic, or specific to a given property, can be also be specified using vw.

The vw graphical interface consistes of a main menu bar, and a collection of "panes", each of which can be brought into view by clicking its tab. These panes are described below.

GETTING STARTED AND EXITING VW

To open a file, if you did not specify one on the command line, either select "Open..." item from the "File" menu to open an existing file, or "New..." to create a new file. New files will be opened in the XEmacs text editor for editing (see section on XEmacs below). If you modify any source files, use the "Reopen" menu item to see these changes in vw. Use "Quit" in the "File" menu to exit vw. If you kill vw with keyboard interrupt, it won't remember its state the next time you run it.

THE BROWSER PANE

Once a file is open, the browser shows a view on the name hierarchy of the SMV program. This includes names of both signals and assertions (properties to be proved). Any name in this hierarchy (such as "foo.bar") is in either an "open" or "closed" state. In the open state, all the children of the name are shown immediately below and indented. In the closed state, the children are not shown. If a given name is closed, and has children that are not shown, then a plus sign (+) appears after the name. Click the plus sign or double-click the name to open it. Click the minus sign, or double-click again to close it.

THE SOURCE PANE

When you click on a name in the name browser, the source pane shows you where in the program source that name is declared. You can also display a source line where the given signal is assigned, or initialized, and if there is a trace in the trace view window, you can display the source line where the signal is assigned at the currently selected time in a trace. You can choose one of these options using the "Show" menu in the source pane.

Note, if the smv file is derived from a verilog file, or another source language, the browser will try to trace the definition back to the original source language file.

The "File" menu in the source pane shows a list of the source files that are currently loaded into SMV (including those that are included using "include" directives.

BROWSING THE SOURCE

When you select the name of a signal or assertion in the browser, the GoTo menu provides to options: "Backward..." and "Forward...". The latter produces a list of all the signals the the selected signal depends on in the current cone. The latter produces a list of all the signals that depend on the current signal in the current cone. When you select an item from the list, you "zap" to that signal. This is just as if you had clicked on the signal name in the name browser. Thus, the signal becomes selected in the browser, and the source view displays the defi nition line of the signal. If necessary, ancestors of the signal name will be "opened" in the browser in order to make the signal name visible. At this point, you can use "GoTo" to see the dependencies of the new signal, and thus work backward or forward through the dependency graph of the program.

The "History" menu contains a short history of the names you have selected in the name browser. Selecting a name in this menu causes you to "zap" to that name.

THE PROPERTIES PANE

A "property" in an smv program is anything that needs to be verifi ed. This includes assertions in temporal logic and refi nement checks. Every property has a name. Assertions are normally named in the program source. An unnamed assertion is given the name of the module instance in which it appears. Refi nement checks have names of the form "x//y", where "x" is a signal name, and "y" is the name of the layer being refi ned.

All the properties in your SMV program are listed in the properties pane. For each, a "status" is given, which can be either "verifi ed", "unverifi ed", or "assumed". A select box allows you to choose to view "All properties", "unverifi ed properties", "Verifi ed properties" or "Assumed properties". A property can be selected for verifi cation by clicking on it in the properties pane. The currently selected property also appears at the bottom of the VW window.

THE CONE PANE

The cone pane shows a list of all the signals that are in the "cone" of the currently selected property. This is the set of all signals on which the property depends, for a given abstraction. For each signal, the cone pane shows the layer or layers whose defi nitions of that signal are used in the abstraction. For an explantion of why a given layer was chosen for a given signal, click on that signal and select "Explain abstraction" from the "Abstraction" menu. The cone pane also shows you how many of the signals in the cone are "state variables" and "combinational" variables. You can choose to see the "Entire cone", "All Variables", "State variables", or "Free variables" (the latter are those variables that are unconstrained in the abstraction).

VERIFYING PROPERTIES

To verify all of the properties in the program, select "Verify" from the "Props" menu. To verify only the selected property, select "Verify property" from the "Props" menu, where "property" is the property name. This will run the smv model checker in the background, and notify you when the verifi cation is completed.

To kill the verifi cation in progress, select "Props:Kill verifi cation".

THE LOG PANE

During verication, a log of the verifi cation process appears in this pane. Assuming you are using the "−v 1" option, you can observe this pane to see the sizes of the BDD’s as the model checking progresses.

THE RESULTS PANE

The results pane shows the results of the most recent verifi cation run. Each property that was checked is listed, along with its truth value. Clicking on a property in the results pane will cause it to become the "current" property, as if it had been selected in the properties pane. In addition, if you click on a false property, a counterexample for it will appear in the trace pane.

THE USING PANE

When a property is selected for verifi cation, the "using" pane displays a list of all the temporal logic assertions that are used as assumptions when proving the current property. For each property, it displays whether the property is assumed "up to t", "up to t - 1", or "always".

THE TRACE PANE

The trace view displays counterexamples and simulation traces. It is essentially a spreadsheet, where the rows represent signals and the columns represent time. Initially, all of the signals in the cone appear in the trace pane. If there is a large number of signals, it is sometimes more convenient to delete all the rows (select Edit|Delete All), and then insert the signals you are interested in. To insert a signal, drag it from the Browser or the Cone pane to a cell in the Trace pane. This will insert a new row in the Trace pane containing the trace of the selected signal. If the signal has children, a new row will be created for each child (thus, for example, of you drag the "top level" into the trace pane, you will all of the available traces.

An alternative method is to select a signal in the browser or cone panes, click a row to insert in, and and choose "Edit|Insert". This will open a new row, and insert the selected signal name in the leftmost column. You can select this signal by just pressing return. Or you can type in the name of a signal and press return. Once you have a set of signals you like in the trace pane, you can save this set to a file by choosing "View|Save Trace View..." in the trace pane. This configuration can be restored by using "View|Restore Trace View...". Note that this saves only the names of the signals, and not the traces themselves, which may have changed when you later restore the trace view.

Normally, SMV returns to a view of all the signals whenever a new trace is displayed. If you don't like this behavior, check "View|Show Requested Traces" in the trace pane.

Use "View|Zoom In" and "View|Zoom Out" to change the column scale of the trace display.

To see where in the source code a given signal was assigned its value at a given time, click on a cell in the trace pane, then switch to the source pane and select "Show|Where Assigned In Trace".

Instead of using the mouse, you can also select cells in the trace pane using the arrow keys on your keyboard, or using "incremental search" (see INCREMENTAL SEARCH below).

MODIFYING TRACES AND SIMULATION

You can treat a trace in the trace view window like a spreadsheet, modifying some values, and then recomputing the entire trace to reflect these modifications. This allows vw to be used as an interactive simulator for smv programs, though a fairly inefficient one.

You can change the value in a cell by clicking on the cell and choosing "Suggest value" or "Override value" from the "Edit" menu. Then type a value into the cell and press return. You can then recalculate the trace by choosing "Run|Recalculate Trace". An "override" causes the given value to be fixed no matter what value is actually assigned by the program to the given signal at the given time. A "suggestion" sets the value of the signal only if the suggested value is among the possible nondeterministic choices for that signal at that time.

To extend the trace, choose "Run|Extend Trace..." and enter the number of columns to add. This will give you a longer trace to work with. You can then enter some values and recalculate.

Hint: if you have asynchronous processes in your model, you can select a processes to run at a given time by using "Override value" to set its "running" variable to one.

LOADING AND SAVING TRACES

The trace curently in the trace view window can be saved to a file with "File|Save Trace". You can load a trace into the trace view window with "File|Load Trace". A new trace, consisting of only an initial state, can be created with "File|New Trace".

SELECTING PROPERTY SPECIFIC OPTIONS

When you choose a property in the properties pane, the "Options..." item in the "Props" menu becomes activated. When you select this, a menu of model checking options is presented. See the *smv(1)* entry for details on these options. The options you select are stored in "fi le.options" for use by future runs of the model checker. Note that your choices here override the default options.

If you have selected property "foo", you can copy the options from property "bar" to "foo" by selecting "Props:Copy options...", and the selecting "bar" from the "Copy options from..." dialog.

EDITING AN ENVIRONMENT ABSTRACTION

When verifying a given property, you can defi ne an abstract view of the program to use when verifying the property. You do this by choosing one or more "layers" to defi ne each signal. This defi nes the "environment" for verifying the given property. Normally these choices are made in the program source with the "using...prove" declaration. However, you can override these choices on a property by property basis using vw.

Once you have chosen a property, you can set the layer to use for any given signal by clicking on that signal and pulling down the "Abstraction" menu. Note that a signal whose abstraction is not set inherits its abstraction from its parent. Thus, you can set the abstraction for an entire module instance by setting the abstraction for the module instance name.

In an smv program without "layers", you have three choices: "(none)", "." and "free". In the fi rst case, you allow smv to choose the abstraction for this signal. In the second case, you are telling smv to use the signal defi nition found in the "implementation" layer, the lowest layer in the refi nement hierarchy. Finally, if you choose "free" then no defi nition will be used for this signal. The signal will be free to take any value at any time.

Note that if you choose "free" for a given signal, you remove a dependency from the property's environment. This means that the size of the property's "cone" can be reduced, and some signals may disappear from the name browser when "fi lter cone" is selected.

If the smv program has defi ned "layers", then the name of each layer will also appear in the "Abstraction" menu. Choosing a given layer will cause the defi nition of the selected signal to be taken from that layer, when verifying the current property. There are three cases when your choice of abstraction may not be used by the model checker:

1) If the given signal is not defi ned in the given layer, the model checker will choose the next lower layer in which the signal is defi ned.

2) If you are verifying refi nement of foo//bar, you cannot use the defi nition of foo in layer bar, since this would be circular reasoning. Similarly, to cannot chose a defi ntion at a higher layer. In this case, smv uses the next layer below bar for the given signal.

3) The given signal may not be in the dependency cone of the property. In this case, it is not used at all. Note that the dependency cone is a function of the chosen abstraction.

Normally, vw shows you abstraction layer choices that the model checker will actually use in the browser pane. Radio buttons in the "View" menu allow you to toggle between showing the "Requested layer" (the one you select) and the "Actual layer" (the one smv will use).

ABSTRACTION FILES

The environment abstraction for a given property is stored in an abstraction fi le (which you can also edit with a text editor). The fi rst time to make an abstraction selection for a given property, you will be asked if you want to create an abstraction for that fi le. This will create a fi le with an name of the form "#####.abs". It is a good idea not to delete these fi les, unless you want to clear all the abstractions.

The abstraction for a given property is considered an "option" and is also copied by "Props:Copy options...". You have the choice in this case to create a new copy of the abstraction fi le or to share the same abstraction fi le between the two properties. In the latter case, changes to the abstraction of one property will affect the other.

If you want to edit an abstraction fi le by hand, you can fi nd the name of the abstraction fi le for a given property by looking in "fi le.options", under the given property name. The abstraction fi le name is the argument of the "-abs" switch.

SAVING AND RESTORING THE ABSTRACTION FILE

If you are editing the abstraction for a given property and want to save it to disk, choose "File:Save abstraction". If you make a mistake and want to get back the saved version use "File:Restore abstraction". Note, however, that running the model checker will cause the abstraction to be saved, and any previously saved fi le will thus be overwritten.

PROPERTY GROUPS

A property group is a collection of properties that behaves exactly like a single property from the point of view of the foregoing discussion. You create and manage property groups using the "group view" window. A list of the currently existing groups is shown in the left-hand pane of this window. When you select a group by clicking on it, the properties it contains are listed in the right-hand pane. To create a new group, click the "New group" button and enter a name for the new group in the dialog box. To add a signal to the currently listed group, select the signal name in the name browser and click the "Add" button. To delete a signal, select the signal name in group view window and click the "Delete button".

Note that to verify a property group, or edit its options and abstraction, you must fi rst select it in the properties pane. This is not done automatically by selecting the group in the "group view" window. Also note that a group marked "verifi ed" in the proeprties pane when all the properties in the groups are verifi ed.

There are two reasons for creating property groups. The fi rst is that you may have a collection of similar properties or refi nements for which you would like to use the same options and abstraction fi le. Since a group is treated as a single property, there is only one option setting and one abstraction fi le for the entire group.

The second reason for creating a group is that for effi ciency reasons, you may want to check all of the properties in the group in the same model checking run. This means that the transition relation and reached state set will be computed only once for the group. To get this effect, select the option "check all properties together" in the options menu (under "Props:Options...").

CONTROLLING VW FROM THE KEYBOARD

Most operations of vw can be controlled from the keyboard. Any visible menu can be "pulled down" by holding down the "Alt" key and typing the underlined letter in the menu name. For example, the "File" menu can be accessed by typing Alt-f. Once the menu is displayed, a menu item can be selected by typing the underlined letter in the menu item. Thus, for example, to open a fi le, type "Alt-f o". Don't use the shift key, even if the letter is capitalized.

Similarly, each pane can be accessed by holding down "Alt" and typing the letter that is underlined on the pane's tab. This also puts the keyboard focus on the selected pane, which means that further keyboard commands are directed to that pane. Pressing the tab key moves the keyboard focus to the next "widget" that is capable of receiving keyboard input. Focus is indicated by a dark outline around the widget. Thus, for example, to change the selection in the cone pane from "Entire cone" to "State variables", first type "Alt-c" to make the cone pane visible, then press tab until the select box with "Entire cone" is visible, then press the "down arrow" key until "State variables" is highlighted, then press Enter.

INCREMENTAL SEARCH

Items in most of the panes of can be selected from the keyboard using "incremental search" (exceptions are the source and log panes). This is very much like the function of the same name in GNU emacs. To use incremental search, put the keyboard focus on the list that you want to search (for example, type Alt-b to put the focus in the browser pane). Then type "C-s" (i.e., Ctrl and s) to start searching, and type a few letters contained in the signal name you are searching for. Vw will outline the first signal in the browser containing that name. As you continue typing characters, vw will move the the outline so that it always indicates the first entry containing the characters you have typed thus far. To see the next matching entry, type "C-s" again, and so forth. To go back to the previous match, press "C-r". To erase a character you have typed in error, press Del. To stop the search and select the currently outlined item, press Enter. To cancel the search taking no action, press "C-g". You also start a search backward from the currently selected entry by typing "C-r". In the browser pane, once you have selected an entry, you can open or close it by pressing Enter again. You can also move up or down in the list using the arrow keys.

USING VW WITH XEMACS

Vw uses the XEmacs version 21.1 as its text editor. To use this feature, you must have XEmacs installed on your system. See www.xemacs.org for instructions on downloading and installing XEmacs. In addition, you must put the line

(load-library "smv-hooks")

in your ~/.emacs file, and the files "smv-hooks.el" and "smv-mode.el" must be in some directory in the list "load-library-path".

Files ending in ".smv" will use smv-mode. This is very similar to c-mode, except that it supports the smv syntax. Further, in smv-mode, a pull-down menu option is provided to run Vw on the current buffer.

Vw uses XEmacs via the gnuserv protocol. To edit a source file, display it in the Source pane, and select the "File|Open in emacs" item in the Source pane menu (not in the main menu). The XEmacs "point" will be moved to the currently highlighted line (for example, the line where a syntax error occurred). After modifying the files, select "File|Reopen" from the main menu. If you have any files in XEmacs that are modified but not saved, you will be asked if you want to save them. The modified sources will then be reread into SMV. When you create a new SMV file with the "File|New" menu item, SMV will also open the file in XEmacs.

Note, SMV is only compatible with XEmacs version 21.1, and not with FSF emacs.

SEE ALSO

smv(1),vl2smv(1)
K.L.**McMillan,**The SMV language
Kenneth L. McMillan, *Symbolic Model Checking,* **Kluwer, 1993.**

BUGS

See the release notes.

AUTHOR

Kenneth L. McMillan, Cadence Berkeley Labs
mcmillan@cadence.com